



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

Seminarausarbeitung  
Andreas Herglotz  
Community Driven Development

Andreas Herglotz  
Community Driven Development

Seminararbeit im Rahmen der Veranstaltung Anwendungen 2  
im Studiengang Informatik (M. Sc.)  
am Studiendepartment Informatik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Professor : Prof. Dr. Kai von Luck

Abgegeben am 29. Februar 2008

**Andreas Herglotz**

**Thema der Seminararbeit**

Community Driven Development

**Stichworte**

Community Driven Development

**Kurzzusammenfassung**

In dieser Ausarbeitung wird zuerst die Bedeutung des Begriffs Community Driven Development geklärt. Der zweite Schwerpunkt liegt auf dem Vergleich aktueller Projekte zu diesem Thema.

**Andreas Herglotz**

**Title of the paper**

Community Driven Development

**Keywords**

Community Driven Development

**Abstract**

This paper has two focuses. The one is to specify what Community Driven Development is. The other one is the comparison of current Community Driven Development projects.

# Inhaltsverzeichnis

<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Zielsetzung . . . . .	1
<b>2 Community Driven Development</b>	<b>2</b>
2.1 Anforderungen . . . . .	2
2.1.1 Web 2.0 Anforderungen . . . . .	2
2.1.2 Weitere Anforderungen . . . . .	4
2.2 Abgrenzung . . . . .	5
2.3 Einteilung . . . . .	5
<b>3 Aktuelle Projekte</b>	<b>6</b>
3.1 Mozilla . . . . .	6
3.2 Wikipedia . . . . .	7
3.3 Second Life . . . . .	7
3.4 Android . . . . .	8
3.5 Weitere Projekte . . . . .	8
3.6 Vergleich der Projekte . . . . .	9
3.7 Herausforderungen . . . . .	10
3.7.1 Vertrauen . . . . .	10
3.7.2 Oszillierender Code . . . . .	10
3.7.3 Community vs. Administration . . . . .	10
<b>4 Zusammenfassung</b>	<b>11</b>
4.1 Fazit . . . . .	11
4.2 Ausblick . . . . .	11
<b>Literaturverzeichnis</b>	<b>12</b>

# 1 Einleitung

In einem ersten Schritt wird auf die Motivation eingegangen sich mit Community Driven Development zu beschäftigen. Darauf aufbauend werden die Ziele dieser Ausarbeitung definiert.

## 1.1 Motivation

Community Driven Development (CDD) bezeichnet ganz allgemein eine Gruppe Gleichgesinnter, die gemeinsam ein Ziel verfolgen, die gemeinsam etwas erschaffen wollen. Dabei sind der Community bei der Kreativität des Schaffens keine Grenzen gesetzt. Es werden ganze Betriebssysteme, Programme, Spiele, Sicherheitsmechanismen, etc. erstellt. Aber es gibt auch Communities, die nicht so informatikaffin sind und durch einfache Schnittstellen virtuelle Welten schaffen oder ein riesiges qualitativ hochwertiges Kompendium schaffen. Dabei gibt es für die einzelnen Mitglieder immer irgendeine Art von Motivation wie beispielsweise Selbstbestätigung. Diese Art von CDD gibt es schon viele Jahre in verschiedenen Ausprägungen. Das besondere in diesem Fall ist, dass CDD in einem gewissen Bereich, nämlich dem Schaffen von Inhalt, auf den Pervasive Gaming Kontext übertragen werden soll.

## 1.2 Zielsetzung

Das Ziel dieser Ausarbeitung ist es zum einen näher zu klären, was CDD ist. Dabei werden verschiedene Bereiche des CDD unterschieden. Zum anderen sollen aktuelle Projekte, die auf CDD bauen, erst vorgestellt werden. In einem weiteren Schritt werden diese verglichen. Zum Schluss werden die Herausforderungen, die es in diesem Bereich gibt, erläutert.

## 2 Community Driven Development

Als geeigneten Einstieg in CDD lässt sich Richard Stallman als Gründer des GNU-Projektes nennen. Er verarbeitete im September 1983 in einer Email sein Vorhaben mit jedem, der gerne möchte, ein neues Unix Betriebssystem zu entwickeln, welches jedem interessierten Nutzer zur Verfügung gestellt werden sollte. Dieses ist ein Meilenstein in der Entstehungsgeschichte von CDD.

Es gibt keine genaue Definition von CDD, daher werden im Folgenden Aspekte, die CDD ausmachen, vorgestellt.

### 2.1 Anforderungen

Es werden erst Anforderungen an CDD im Web 2.0 Zusammenhang gestellt und dann weitere allgemeine Anforderungen definiert.

#### 2.1.1 Web 2.0 Anforderungen

Die folgende Abbildung [2.1](#), in der Tim O'Reilly eigentlich das Web 2.0 beschreibt, spiegelt viele wichtige Aspekte von CDD wider. Als Beispiele führt er im Web 2.0 Kontext Flickr oder Wikipedia auf, die zweifellos auch eine Art von CDD sind. Es werden die wichtigsten Punkte kurz näher erläutert.

„**Control over unique, hard-to-recreate data sources that get richer as more people use them**“: Damit ist beispielsweise gemeint, dass nicht das Kartenmaterial an sich sondern diejenigen speziellen schwer reproduzierbaren Zusatzinformationen, die durch die Community entstehen, das wichtige sind. Je größer die Community ist, die an der Erweiterung aktiv teilnimmt, desto wertvoller wird das Material. Das Kartenmaterial von den Trailblazern oder TomTom Go kann hier exemplarisch angeführt werden. Auf diese beiden Projekte wird im Kapitel [3.5](#) etwas näher eingegangen.

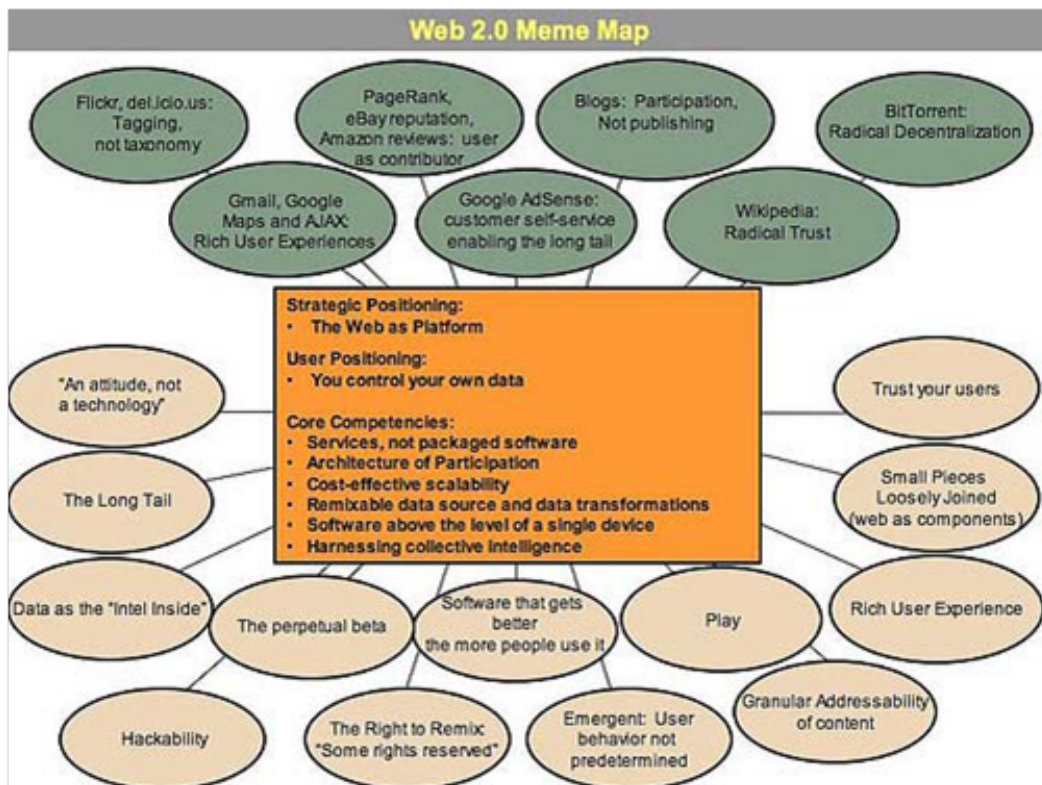


Abbildung 2.1: Tim O'Reillys Web 2.0 Meme Map

„**Trusting users as co-developers**“ Der Nutzer von Open Source Software kann, wenn er über entsprechende Fähigkeiten verfügt mit in die Entwickler-Community einsteigen und das Produkt weiter vorantreiben. Als Beispiel lässt sich Mozilla erwähnen (siehe Kapitel 3.1). Eine solche Zusammenarbeit funktioniert nur, wenn ein gewisses Maß an Vertrauen vorhanden ist.

„**Harnessing collective intelligence**“ Dieser Aspekt lässt sich am Beispiel von Wikipedia sehr gut erklären. Die Intelligenz der Gemeinschaft bzw. der Community ist sehr groß. Es weiß nicht jeder alles, aber bestimmte Leute haben spezielles Wissen, von dem alle profitieren sollen. Wikipedia spiegelt dies gut wider. Es ist durch die Community, die Artikel geschrieben hat, eine unglaublich breite und teilweise auch spezifische Menge an Wissen in diesem Kompendium abgebildet worden.

„**Leveraging the long tail through customer self-service**“ Mit „the long tail“ im Bezug auf Web 2.0 sind die kleinen Webseiten gemeint, die den Hauptanteil im Internet ausmachen, verglichen mit den wenigen Webseiten der großen Firmen. Bezogen auf CDD sind damit alle

gemeint, die in irgendeiner Art und Weise aktiv entwickeln oder Inhalt schaffen wollen. Das ist die breite Masse, die Community, die insgesamt ein viel größeres Gewicht einnehmen kann als die wenigen Firmen, die sich damit beschäftigen. Eben dieser Aspekt soll sich zum Vorteil gemacht werden. Wikipedia stellt dies Phänomen wieder gut dar. Bei einer Firma, die ein Lexikon erstellt, gibt es eine kleine Zahl von Mitarbeitern, die die notwendigen Informationen zusammentragen. Bei Wikipedia ist es eine unglaublich große Anzahl von Benutzern, die sich der Aufgabe gestellt haben. Man sieht an diesem Beispiel welchen Nutzen aus „the long tail“ gezogen werden kann.

**„Lightweight user interfaces, development models, AND business models“** Ein wichtiger Erfolgsfaktor für CDD ist, dass für die entsprechende Community und deren Fähigkeiten entsprechende Schnittstellen geschaffen werden. So ist für die Entwicklung von Firefox eine vernünftige Programmiersprache notwendig, für das Erstellen von Artikeln bei Wikipedia hingegen eine möglichst einfache Benutzerschnittstelle, damit auch nicht informatikaffine Benutzer an der Community aktiv teilnehmen können. Einfache Modelle sind wichtig, um der Community die Möglichkeit zu geben ihre Kreativität auszuleben und neue Komponenten zu schaffen. Die Barrieren für diese kreative Arbeit müssen dazu möglichst gering sein.

## 2.1.2 Weitere Anforderungen

Unter weiteren Anforderungen werde im Wesentlichen die folgenden zwei verstanden.

**Ownership/Copyright** Dies sind Fragen, um die sich zweifellos gekümmert werden muss. Dabei geht es zum einen darum nicht in illegale Bereiche zu kommen, weil jemand aus der Community zum Beispiel beim Erstellen von Inhalt das Copyright ausversehen oder absichtlich verletzt hat. Ownership hingegen behandelt das Problem, wem der geschaffene Quellcode oder Inhalt gehört und wie dieser weitergenutzt werden darf. Das ist ein für die Community besonders wichtiger Faktor, da zum einen sehr ungern etwas für die Allgemeinheit erstellt wird, wenn man danach keine Rechte mehr daran hat und zum anderen wenn man etwas erstellt und dies kostenlos zur Verfügung stellt, möchte man nicht unbedingt, dass jemand anderes genau dies für kommerziellen Einsatz missbraucht. Es gibt viele Lizenzen in diesem Bereich. Für eine detaillierte Betrachtung dieser Lizenzen ist es sinnvoll einen Anwalt zu rate zu ziehen, da man sich sehr schnell aus dem informatikspezifischen Bereich entfernt.

**Qualitätssicherung** Die Qualitätssicherung ist sowohl bei der Softwareentwicklung als auch beim Erschaffen von Inhalt notwendig, damit die Software stabil und sicher läuft und der Inhalt richtig ist und keine Gesetze verletzt. Bei diesem Punkt befindet man sich schnell im



Bereich der Administration. Es muss besondere Mitglieder der Community geben, die mehr Rechte haben als andere, um solche Dinge zu klären.

## 2.2 Abgrenzung

Dieser Abschnitt spricht kurz Bereiche an, die für CDD zwar wichtig sind, aber in diesem Zusammenhang nicht weiter betrachtet werden. CDD baut auf dem Konzept von verteilten Systemen auf und lässt sich ohne diese nicht realisieren. Dennoch liegt die Entwicklung von Tools für verteilte Systeme nicht im Fokus dieser Arbeit. Des Weiteren werden allgemeine Themen wie Sicherheit, verteilte Transaktionen (Techniken und Konzepte), etc. nicht behandelt.

## 2.3 Einteilung

Man kann CDD in vier Gruppen unterteilen wobei die vierte Gruppe nur auf Grund der Vollständigkeit aufgeführt wird.

1. **Wizard:** Diese Gruppe der Experten ist zumeist die kleinste, da in diesem Bereich schon sehr ausgeprägtes Informatikwissen zu den Grundlagen gehört. Sehr häufig schafft diese Gruppe keinen Inhalt, sondern entwickelt gemeinsam Open Source Software in entsprechenden Programmiersprachen.
2. **Programmer:** Die Programmierer nutzen meistens Scriptsprachen. Diese sind in ihrer Ausdrucksstärke im Vergleich zu den Experten schon stark eingeschränkt. Diese Gruppe befindet sich in der Art der Beteiligung zwischen den Experten und den Autoren. Häufig ist dies eine versierte Beteiligung an dem Schaffen von Inhalt, da der Gruppe durch ihre Kenntnisse mehr Möglichkeiten zur Verfügung stehen als den Autoren.
3. **Author:** Dies ist bzw. soll die größte Gruppe sein, die sich aktiv am CDD beteiligt. Charakterisiert wird sie durch nicht zwangsläufig informatikaffine Benutzer, die einfache Schnittstellen benutzen wie beispielsweise einen Editor oder eine graphische Benutzeroberfläche. Dadurch ist die Ausdrucksstärke stark begrenzt. Zumeist schafft diese Gruppe Inhalt in verschiedensten Formen.
4. **Consumer:** Diese Gruppe wird nicht weiter betrachtet, da diese die Applikation einfach nur benutzt und nicht aktiv daran mitwirkt. Allerdings ist es sinnvoll möglichst viele aus dieser Gruppe dazu zu bewegen sich der dritten Gruppe zuzuordnen.

## 3 Aktuelle Projekte

Dieses Kapitel setzt sich mit verschiedenen aktuellen Projekten zum Thema Community Driven Development auseinander.

### 3.1 Mozilla

Die 1998 von Netscape gegründete Organisation Mozilla.org hat es sich zum Ziel gemacht kostenlose Software zu entwickeln. Das bekannteste Produkt ist der Browser Firefox. Um Software kostenlos zur Verfügung stellen und diese auch kontinuierlich weiter entwickeln zu können, wurde der offene Quellcode damals für jeden Interessierten zum Download bereit gestellt. So entwickelte sich eine Community rund um den erfolgreichen Browser. Im Jahr 2003 wurde die zwischenzeitlich von AOL aufgekaufte Entwicklungsabteilung geschlossen und die als gemeinnützig anerkannte Mozilla-Stiftung gegründet. 2005 wurde die Mozilla Corporation für alle kommerziellen Produkte gegründet, um die Stiftung nicht zu gefährden. Bei der Community handelt es sich zum einen um Entwickler und zum anderen gibt es mittlerweile auch ein Tool namens Bugzilla, mit dem die Benutzer der Mozilla Produkte automatisch aktiv an der Verbesserung Software teilnehmen können. Dies Tool meldet automatisch Fehler an die Programmierer.

**Herausforderungen** Bei Mozilla gibt es zwei wesentliche Herausforderungen. Die eine ist durch genügend Spenden und einen entsprechenden Finanzhaushalt dafür zu sorgen nicht von Firmen abhängig oder gar aufgekauft zu werden. Der andere sehr wichtige Faktor ist, dass genug Benutzer mit dem Browser erreicht werden, damit Mozilla genug Einfluss hat, damit möglichst alle Webseiten im Firefox funktionieren. Ein gewisser Marktanteil ist einfach für ein standard-basiertes Internet notwendig.

Sehr interessant im Zusammenhang mit Android ist die immer wichtiger werdende mobile Nutzung von Browsern. Denn durch die hinzugewonnene Mobilität ist nicht nur bekannt, was von wem gesucht wird, sondern auch noch zusätzlich die Information über den aktuellen Aufenthaltsort. Dadurch entstehen ganz neue Chancen aber auch Risiken bezüglich der Werbemöglichkeiten und des Datenschutzes. In diesem Zusammenhang kann der offene

Quellcode, der von einer offenen Community und nicht von einer umsatzgeprägten Firma entwickelt wurde, entscheidend bei der Auswahl des mobilen Browsers mitwirken.

## 3.2 Wikipedia

Ein nicht kommerzielles Projekt, bei dem es darum geht gemeinsam mit dem speziellen Fachwissen des Einzelnen oder dem ergänzenden Wissen Mehrerer ein Lexikon zu erstellen. Diese Community basiert zum einen auf Vertrauen und zum anderen auf Objektivität. Da dies alleine nicht ausreicht, gibt es eine Administrationsschicht. Diese besteht aus der obersten Autorität zum Schlichten, dem Gründer Jimmy Wales (benevolent dictator). Darunter befindet sich ein von den Teilnehmern gewähltes Schiedsgericht (arbitration committee). Zu guter letzt gibt es noch einfache Administratoren.

Da dies Projekt wie anfangs erwähnt nicht kommerziell ist und natürlich Kosten entstehen, müssen diese gedeckt werden. Die Finanzierung ist vollständig über Spenden von Privatpersonen oder Unternehmen realisiert.

**Herausforderungen** Bei Wikipedia gibt es drei Hauptprobleme. Das erste ist das gefährliche Halbwissen, da es in der Gesellschaft immer nur eine Minderheit gibt, die über entsprechendes Fachwissen verfügt. Dabei läuft die Minderheit stetig Gefahr von der Mehrheit „korrigiert“ zu werden. Dadurch würde die Brauchbarkeit des Lexikons stark reduziert, wenn das nicht reguliert würde. Schlimmer ist in dem Zusammenhang noch die absichtliche Manipulation von Artikeln. Daher ist die Administrationsschicht unumgänglich, die Artikel oder Benutzer sperren können. Gleichzeitig müssen auch Urheberrechte bei den veröffentlichten Artikeln gewahrt werden, damit es diesbezüglich nicht zu Klagen kommt. Bei all den Problemen darf der administrative Overhead nicht übertrieben werden, da sonst die Gefahr besteht, dass die Community daran kaputt geht.

## 3.3 Second Life

Ist ein kommerzielles Produkt, indem virtuelle Welten/Leben geschaffen werden. Es wird also mit verschiedenen Mitteln Inhalt erstellt. Direkt in Second Life gibt es ein 3D Modellierungstool für die einfachen Benutzer. Für versiertere Benutzer gibt es eine eigens dafür geschaffene Linden Scripting Language. Dabei steigt der Anspruch für den Nutzer, aber auch die Möglichkeiten Inhalt zu erschaffen. Innerhalb dieser virtuellen Welt kann man handeln, Dinge verkaufen und mit anderen kommunizieren.

**Herausforderungen** Es gab es bisher verschiedene Probleme. Die Sever waren den Anforderungen nicht gewachsen, so dass es enorme Probleme mit der auftretenden Last gab und die Latenzzeiten zu stark anstiegen. Des Weiteren gibt es moralische Probleme. Ein konkreter Fall ist die Pornographie in Second Life. Nicht zu vergessen sind auch Probleme, die die Legalität im Bezug auf den Lindendollar betreffen.

### 3.4 Android

Android ist eine Open Source Handyplattform mit frei verfügbarem SDK. Damit hat Google im Bereich der mobilen Endgeräte eine gute Möglichkeit geschaffen CDD voranzutreiben. In diesem Zusammenhang ist erst einmal die Softwareentwicklung gemeint, die durch die Community stark vorangetrieben werden kann. Darauf aufbauend sind auch Szenarien wie der Pervasive Community Messenger [Herglotz \(2008\)](#) denkbar, bei dem die Community jederzeit und überall Inhalt erschaffen kann. Das die Nutzergemeinschaft ein starkes Interesse daran hat selbst mitzuwirken zeigt die Entscheidung von Apple. Diese haben sich entgegen der eigentlichen Firmenphilosophie dahingehend entschieden doch offene Schnittstellen für das iPhone zur Verfügung zu stellen, um die Kunden nicht an Android zu verlieren.

**Herausforderungen** Ob sich eine große Community in Kontext von Android entwickelt hängt im Wesentlichen von zwei Dingen ab. Das eine ist die Lizenz, unter der die Software entwickelt wird. Der andere ist die Intention von Google möglichst platzierte und effektive Werbung zu schalten. Die Zeit wird zeigen wie kritisch die Community dies beurteilt.

### 3.5 Weitere Projekte

In diesem Zusammenhang lässt sich noch das Projekt „Trailblazers“ der HAW Hamburg anführen ([Sven Stegelmeier \(2008\)](#)). Dabei geht es um das erstellen von Kartenmaterial mit Hilfe mobiler Smart-Clients - insbesondere für gehbehinderte Menschen.

Ähnlich dazu verhält sich das neue Navigationssystem TomTom Go. Dabei geht es ebenfalls um das Anpassen von Kartenmaterial an den Alltag. Wenn eine Straße gesperrt ist, teilt man dem Navi dies mit. Beim Synchronisieren zu Hause wird dies den anderen Nutzern mitgeteilt.

Die Community in beiden Fällen sind die Nutzer, die relativ automatisch Kartenmaterial erschaffen bzw. verändern. Dabei ist bereits eine Bewegung weg vom rein stationären CDD vor dem PC hin zu mobilem CDD zu erkennen.

## 3.6 Vergleich der Projekte

In der nachfolgenden Tabelle werden die aktuellen Projekte gegenübergestellt.

	Mozilla	Wikipedia	Second Life	Android
Community	Wizards	Authors	Authors	Wizards
Quality-assurance	Community	Community, Admins	Second Life owner	(Developer)
Motivation	Self-Expression	Self-Expression	Fun	Self-Expression
Participation	Software development	Writing articles	Imitation of life	Software development
Subsecond Response	Multiple cycles	Adhoc	Adhoc	Unknwon
Ownership/ Copyright	GPL	GFDL	Copy, Mod, Trans	ASL

Abbildung 3.1: Vergleich der aktuellen Projekte

Betrachtet man die Community so ist zu erkennen, dass gerade im Bereich der Softwareentwicklung (Mozilla, Android) die Ansprüche ziemlich hoch sind (Wizards). Sobald man Inhalt schaffen kann (Wikipedia, Second Life) sind die Barrieren, als nicht informatikaffine Person (Authors) daran aktiv mitwirken zu können, deutlich geringer. In diesen Bereichen kann sich im Grunde jeder beteiligen. Durch Tools, wie das in Kapitel 3.1 angesprochene Bugzilla, werden Möglichkeiten geschaffen die Community zu vergrößern.

Uner dem Punkt Qualityassurance ist deutlich zu erkennen, dass es immer irgendeine Instanz gibt, die dies sicherstellt. In diesem Zusammenhang muss noch einmal erwähnt werden, dass die Community und die Administration gegenläufige Bewegungen sind. Dazu mehr in Abschnitt 3.7.3.

Die Motivation ist bei jedem verschieden. Häufig ist dies auf Selbstdarstellung zurückzuführen. Bei Second Life ist es eher der Spieltrieb. Denkbar sind aber auch Belohnungs-/Bestrafungssysteme.

Die Möglichkeiten sich in die Community einzubringen sind vom Projekt abhängig. Bei Mozilla ist es in der Hauptsache die Softwareentwicklung, bei Wikipedia das Erstellen von Artikeln. Des Weiteren wird unterschieden zwischen den Reaktionszeiten bis die Änderungen sichtbar sind. Bei Wikipedia und Second Life passiert dies adhoc, so wie man es als Benutzer auch erwartet. Beim Software entwickeln ist dies anders. Hier werden mehrere Zyklen durchlaufen, um die Qualität und Stabilität der Software zu gewährleisten.

Als letzter Vergleichspunkt ist Copyright und Ownership anzuführen. Es gibt verschiedenste Lizenzen in dem Bereich. Auf diese soll hier nicht näher eingegangen werden, da dies eher ein juristisches Problem ist.

## 3.7 Herausforderungen

Dieser Abschnitt beschäftigt sich noch einmal mit den Herausforderungen die für CDD im Allgemeinen gelten und noch zu lösen sind.

### 3.7.1 Vertrauen

Dies ist ein sehr wichtiger Punkt. Hier steht das Vertrauen der Community zu dem Produkt, da der Quellcode offen ist und man daran mitwirken kann, wenn man möchte, im starken Gegensatz zu den Firmen. Die Firmen haben vermutlich das gleiche Vertrauen, aber in dem Fall, dass etwas nicht so läuft wie geplant, keine rechtliche Handhabe.

### 3.7.2 Oszillierender Code

Es geht darum eine Möglichkeit zu schaffen mit diesem Problem umzugehen oder am besten es zu lösen. Zuerst muss dies Problem überhaupt erkannt werden. Ein Lösungsansatz ist im CVS die Häufigkeit von geänderten Bereichen zu überprüfen. Sind die Änderungen an manchen Stellen auffällig hoch, so müssen diese genauer beobachtet werden. Ziel ist natürlich eine vollkommen automatische Lösung.

### 3.7.3 Community vs. Administration

Die Schwierigkeit ist, dass große Projekte ohne Administration aus dem Ruder laufen. Genau an dieser Stelle ist das Risiko groß durch zu viel Administration die Community zu zerstören oder durch zu wenig Administration nicht das Notwendige zu erreichen. Da scheint die einzige Lösung ein Mittelmaß zu finden.

## **4 Zusammenfassung**

Abschließend wird ein kurzes Fazit gegeben und ein Ausblick auf die Masterarbeit.

### **4.1 Fazit**

Nachdem ein Verständnis von CDD geprägt wurde und verschiedene Projekte miteinander verglichen wurden, wird schnell klar, welches großes Potential dahinter steckt. Dennoch gibt es eine Menge von Problemen und Herausforderungen in diesem Bereich.

### **4.2 Ausblick**

Das große Potential von CDD im Desktop-Bereich wurde herausgestellt. Durch Trailblazers, TomTom Go und natürlich Android scheint eine sehr interessante Weiterentwicklung möglich zu sein. Daher ist es das Ziel CDD im Rahmen von Pervasive Gaming während der Masterarbeit umzusetzen. Es soll ein Pervasive Community Messenger erstellt werden, der es ermöglicht überall und zu jeder Zeit Inhalt von einem mobilen Gerät zu erschaffen.

# Literaturverzeichnis

- [Herglotz 2008] HERGLOTZ, Andreas: *Community Driven Development and Social Networking Becomes Pervasive*, HAW Hamburg, Seminararbeit, 2008
- [Online 2008] ONLINE, Spiegel: „Die Firefox-Gemeinschaft kann niemand kaufen“. 2008. – URL  
<http://www.spiegel.de/netzwelt/web/0,1518,534938,00.html>. –  
Abrufdatum 28.02.2008
- [O'Reilly 2005] O'REILLY, Tim: *What Is Web 2.0? Design Patterns and Business Models for the Next Generation of Software*. [www.oreilly.com](http://www.oreilly.com). September 2005
- [Stallman 1983] STALLMAN, Richard: *GNU Operating System*. 1983. – URL  
[www.gnu.org/gnu/initial-announcement.html](http://www.gnu.org/gnu/initial-announcement.html). – Abrufdatum 28.02.2008
- [Stearns u. a. Jan. 2006] STEARNS, H. ; GARGUS, J. ; SCHUETZE, M. ; LOMBARDI, J.: Simplified Distributed Authoring Via Component-based Object Construction and Deconstruction in Collaborative Croquet Spaces. In: *Creating, Connecting and Collaborating through Computing, 2006. C5 '06. The Fourth International Conference on* (Jan. 2006), S. 79–87
- [Sven Stegelmeier 2008] SVEN STEGELMEIER, Kai von L.: *The Anatomy of the Trailblazers System*, HAW Hamburg, Projektausarbeitung, 2008
- [Wikipedia 2006] WIKIPEDIA: *Wikipedia Organisation*. 2006. – URL  
<http://de.wikipedia.org/wiki/Wikipedia#Organisation>. –  
Abrufdatum 28.02.2008
- [Wikipedia 2008] WIKIPEDIA: *Second Life*. 2008. – URL [http://en.wikipedia.org/wiki/Second\\_Life#Technical\\_information](http://en.wikipedia.org/wiki/Second_Life#Technical_information). –  
Abrufdatum 28.02.2008